

Perancangan dan Pembuatan Aplikasi Resep Makanan Berbasis Android

Paul Agustinus¹, Agustinus Noertjahyana², Justinus Andjarwirawan³

Program Studi Teknik Informatika Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121 -131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) - 8417658

cool_helios@yahoo.com¹, Agust@petra.ac.id², Justin@petra.ac.id³

Abstrak

Penggunaan *device* berbasis *Android* dari tahun ke tahun mengalami peningkatan yang sangat pesat. Salah satu contohnya adalah aplikasi pada bidang kuliner. Makanan merupakan hal yang bisa dikatakan sangat penting dalam kehidupan sehari-hari, namun faktanya banyak orang di Indonesia tidak bisa memasak.

Para pengguna *device* *Android* bisa mendapatkan aplikasi yang memudahkan untuk memasak makanan cita rasa Indonesia dengan mudah. Bagaimanakah sebuah aplikasi resep makanan berbasis *Android* dapat mempermudah individu untuk melakukan pemilihan makanan Indonesia berdasarkan kategori tipe makanan, bahan makanan utama, serta lama waktu memasak. Aplikasi yang dikembangkan berbasis *Android* menggunakan bantuan *yahoo query language* untuk mengkonversi data mentah menjadi data yang bisa diolah. Perintah dijalankan dengan perintah *PHP* dan *client* melakukan request kepada *webserver* sehingga menerima *JSON*.

Aplikasi ini merupakan sebuah aplikasi yang dapat memberi kemudahan kepada *Android user* untuk mendapatkan resep masakan Indonesia dengan mudah. Hasil kuesioner pengguna aplikasi dari 20 orang responden ini mulai dari yang tertinggi adalah 90% berkata bahwa antarmuka memiliki tampilan yang bagus, 75% berkata bahwa aplikasi ini mudah dipakai, 70% berkata bahwa aplikasi ini memiliki kelengkapan fitur yang cukup, dan 70% berkata bahwa aplikasi ini berguna bagi kehidupan sehari-hari.

Kategori dan Deskripsi Subjek

D.3.3 [Java]: Language Constructs and Features – Abstract Data Type, Classes and Object, Data Type and Structures.

Istilah Umum

Algoritma, Eksperimen.

Kata Kunci

Android, Resep, Filtering, Sumber Data

1. PENDAHULUAN

Penggunaan *device* berbasis *Android* dari tahun ke tahun mengalami peningkatan yang sangat pesat. Salah satu alasan *device* berbasis *Android* banyak digunakan adalah harganya terjangkau dan kegunaan bisa semaksimal mungkin

dikarenakan banyak sekali aplikasi-aplikasi yang disediakan oleh *play store* untuk para pemakai *device* berbasis *Android*.

Salah satu contohnya adalah aplikasi pada bidang kuliner. Makanan merupakan hal yang bisa dikatakan sangat penting dalam kehidupan sehari-hari. Faktanya banyak orang di Indonesia tidak bisa memasak. Menggunakan *device* berbasis *Android*, para pengguna *device* bisa mendapatkan aplikasi yang memudahkan untuk memasak makanan cita rasa Indonesia dengan mudah.

Aplikasi kuliner mengenai resep makanan cita rasa Indonesia sudah menyajikan informasi mengenai berbagai macam resep-resep makanan Indonesia dengan cukup lengkap, selain itu ada juga fitur yang dapat membantu individu untuk mencatat bahan-bahan makanan dari resep yang ada.

Aplikasi resep makanan berbasis *Android* dapat mempermudah individu untuk melakukan pemilihan makanan Indonesia berdasarkan kategori tipe makanan, bahan makanan utama, serta lama waktu memasak. Aplikasi yang dapat memenuhi kemudahan pengguna *device*, kemudahan yang diperlukan dapat ditemukan hanya dengan *one stop clicking*. Oleh karena itu, skripsi ini dibuat untuk dapat mengoptimalkan masalah yang telah dipaparkan diatas.

2. STRUKTUR FILE

Beberapa bagian dalam proses masuk dan keluarnya data terdiri dari beberapa aspek. Aspek tersebut diantaranya:

- pengambilan data dari *source RSS*.
- *Scheduling task*.
- *Request* data dari *webserver*.

Data berasal dari sumber yang telah ditentukan yang tentunya sumber yang menyediakan *RSS*.

2.1. Grab data

Proses pengambilan data memerlukan pengubahan data mentah menjadi data setengah jadi untuk diolah menjadi data yang siap pakai. Pengambilan data memerlukan *YQL*[3] yang merupakan bahasa ekspresif seperti *SQL* yang memungkinkan Anda query, filter, dan bergabung data melalui layanan Web.

Penggunaan *YQL* mendukung aplikasi berjalan dengan sedikit baris. Ketika *query* tersebut diproses, *YQL*[8] *Web Service* mengakses *datasources* dari internet, mentransformasikan data, dan mengembalikan dalam bentuk *XML* atau *JSON*[6] format yang merupakan *lightweight data-*

interchange format dan Sangat mudah bagi manusia untuk membaca dan menulisnya. YQL dapat mengakses layanan di internet bahwa *output* data dalam format berikut: HTML, XML, JSON, RSS, Atom, dan microformat.

2.2. Scheduling Task

Scheduled task adalah perintah yang menjalankan PHPscript secara berkala yang memiliki perintah untuk melakukan pengambilan data secara berkala. *Scheduled task* menggunakan salah satu tool yang bernama *crontab/cronjob*. Cron adalah Unix, Solaris utilitas yang memungkinkan *tasks* yang akan otomatis berjalan di belakang layar secara berkala oleh *daemon* cron.

Task ini sering disebut sebagai pekerjaan cron di Unix, Solaris. Crontab[4] adalah file yang berisi jadwal entri cron yang akan dijalankan dan pada waktu yang ditentukan. Penggunaan crontab dapat dijalankan bilamana nama file ada pada /usr/lib/cron/cron.allow dan nama file tidak ada pada /usr/lib/cron/cron.deny. jika hanya ada cron.deny dan tidak ada isinya, semua *user* dapat menggunakan crontab. Jika file tersebut tidak ada, hanya *userroot* dapat menggunakan crontab. The *allow / deny* file terdiri dari satu nama pengguna per baris.

2.3. Request data

Request data kepada Apache *webserver*[2] merupakan proses permintaan dari sisi Android[7] *client* untuk mendapatkan sebuah nilai. *Client* mengirimkan *request* kepada *webserver* dan *webserver* mengembalikan nilai kepada Android *client*[3] sehingga *client* dapat mengolah data yang sudah diterima untuk ditampilkan pada layar aplikasi.

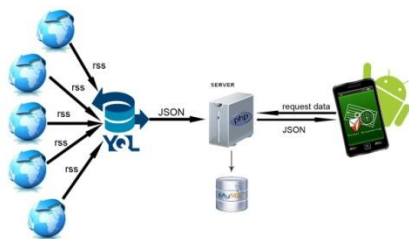
3. ANALISA DAN DESAIN SISTEM

Pada bagian awal Apache *Web server* memiliki beberapa komponen inti yang mendukung mulai dari pengambilan *content*, penyaringan isi konten, pengisian database, dan menyediakan *Web Services*. Komponen-komponen yang dibutuhkan adalah :

- Pengambil data mentah(server).
- Penerimaan data client.
- Blok arsitektur.

3.1. Blok Arsitektur

Blok arsitektur merupakan gambaran umum dalam system ini.



Gambar 1 Blok Arsitektur

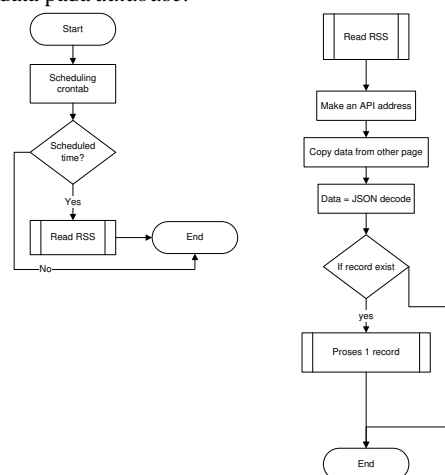
Blok arsitektur dimulai dari pengambilan data oleh PHPscript yang yang berisikan YQL dan pola struktur data *source* yang bersangkutan. Setiap data dipecah menjadi data yang yang dibutuhkan dan dimasukkan kedalam MySQL.

Data yang sudah ada memungkinkan client melakukan *request* pada *webserver* dan *webserver* mengembalikan *request* data berupa JSON format. Client mendapatkan nilai yang diinginkan oleh *client* dan data JSON format dibaca lalu dimasukkan dalam *variable* untuk ditampilkan di layar aplikasi.

3.2. Pengambilan Data Mentah

Proses pengambilan data membutuhkan beberapa komponen yang mendukung dalam bahasa PHP sehingga mampu mengambil data mentah dari RSS yang tersedia, melakukan pemisahan setiap *value*, melakukan pengecekan *value* yang sudah tersedia, memasukkan data ke tabel sementara. Berikut merupakan komponen beserta fungsi-fungsi dari *activity* ini:

YQL untuk menghasilkan format JSON dari RSS yang tersedia, "*curl_init*" untuk mengambil data dari halaman lain, *Strips_tag* untuk menghilangkan tag, "*strpos*" dan *substr* untuk membatasi *code* dalam *browser*, "*explode*" untuk memisahkan setiap *value* yang dibutuhkan, "MySQL" untuk *insert* data pada *database*.

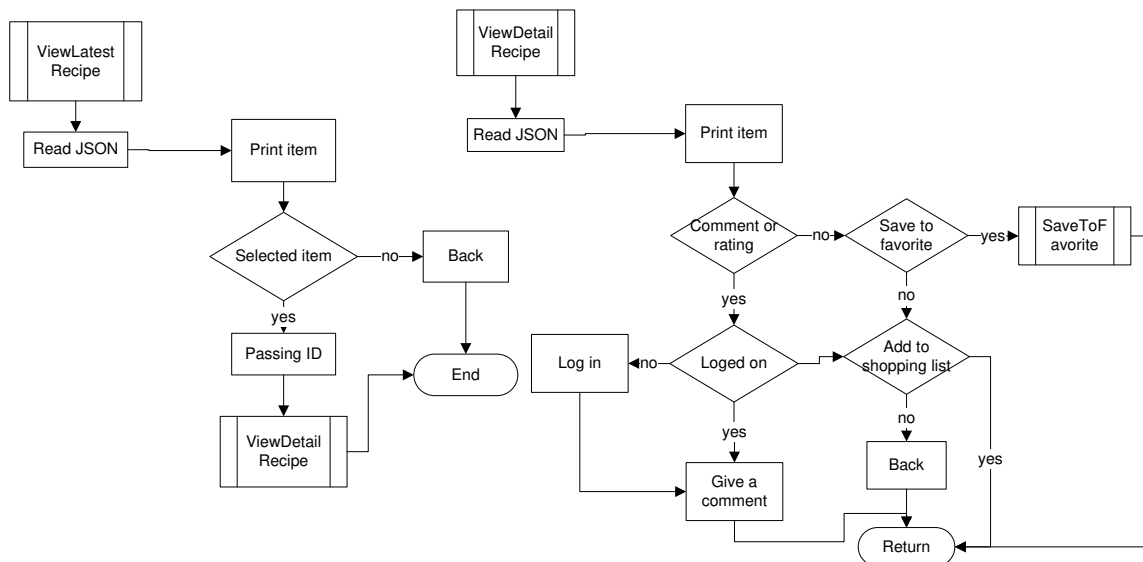


Gambar 2 Proses pengambilan data otomatis

Proses pengambilan data dimulai dari memulai melakukan *scheduling* terhadap perintah untuk mengambil data. Setiap *schedule* yang sudah tercatat, akan menjalankan PHPscript yang sudah dituju untuk melakukan subproses yang bernama *read RSS*. Read RSS dimulai dengan membuat sebuah alamat API dan disalin pada halaman web yang telah disediakan yang berisi hasil *decode* data JSON sehingga dapat diproses setiap satuan data yang tersedia menjadi data yang lebih detail lagi.

3.3. Penerimaan Data Client

Penerimaan data siap pakai merupakan proses dimana *client* menerima data berupa JSON



Gambar 3 Flowchart proses penerimaan data client

Berawal dari menerima data berupa JSON dan data dipisah menjadi sejumlah bagian data lalu data dicetak pada layar aplikasi. Setiap data ditampilkan dalam *listview* yang ada dan setiap *listview* memiliki nilai untuk melakukan request data detail pada webserver. Pada sisi *webserver* menerima *request* dan mengirim data JSON kepada *client*.

4. IMPLEMENTASI

Pengambilan data memiliki dua komponen yaitu *crontab* dan *PHPscript* yang berisi perintah untuk mengambil resep makanan dari sumber data sampai pada *database* yang ada pada *web server*. *Crontab* merupakan sebuah *tool* yang disediakan oleh *web server* untuk melakukan *scheduling*.

Pseudocode 1. Fungsi penulisan *scheduling* dan running scheduled task

```

writeSCHEDULE
if SCHEDULED then RUN PHP
  
```

Dari hasil pseudocode 1 maka akan menjalankan *PHPscript* yang berisikan pengecekan data apakah data ada atau tidak.

Pseudocode 2. Proses pengecekan file.

```

int i = 0;
while (i < total data){
  if(file not exist)
  {
    Explode data, Insert data
  }
  Else
  {
    end
  }
  i++;
}
  
```

Hasil pseudocode 2 jika judul tidak ditemukan dalam *database*, maka *PHPscript* melakukan proses pemecahan data dengan *substr* dan *strpos* berdasarkan pola yang berbeda

untuk setiap *source* RSS yang sudah tersedia, proses pemecahan setiap bahan dan cara pengolahan dapat dipisah dengan cara yang sama namun memiliki pola yang berbeda-beda karena format setiap *website* memiliki pola yang berbeda. Data yang terpecah akan dimasukkan dalam *database*.

Data sudah tersimpan dalam *database server* dan *webserver* menyediakan *webservice* untuk melayani *client* sehingga tidak membebani *client* untuk bekerja terlalu berat. Dalam *webservice* tersedia fungsi-fungsi antara lain :

function view_recipe(), view_detail_recipe(), filter_by_bahan(), filter_by_kategori(), add_komentar(), get_komentar(), upload(), approve(), log_in(), view_my_recipe(), sign_up(), access_token(), view_latest_temp(), view_detail_temp_recipe, add_rating().

Pseudocode 3. Fungsi untuk menaampilkan nama

```

int i = 0; int idm;
while (i < total data){
  if(file exist)
  {
    Return json_encode
    Get idm
  }
  Else
  {
    Return 0
  }
  i++;
}
If(Pressed) Send idm
  
```

Function *view_recipe()* sedikit memiliki persamaan dengan function *view_latest temp()* dan function *view_my_recipe()*. Letak perbedaan dua fungsi ini adalah *view_recipe()* berhubungan langsung dengan tabel data *real* sedangkan function *view_latest_temp_recipe()* berhubungan dengan table *temp* dan keduanya menggunakan

SQL *query* untuk mendapatkan data dan dimasukkan array dan mengembalikan JSON encode dari hasil JSON

Pseudocode 4. Fungsi untuk menampilkan detail

```
Retreive int idm;  
Select data where id is idm from data  
DownloadFileTask()  
Then retrieve data_detail with JSON  
format
```

Function `view_detail_recipe()` memiliki kesamaan dengan function `view_detail_temp_recipe()`. Letak perbedaan dua fungsi ini adalah function `view_detail_recipe()` berhubungan langsung dengan tabel data *realsed* sedangkan function `view_detail_temp_recipe()` berhubungan dengan table *temp* dan keduanya menggunakan SQL *query* untuk mendapatkan data dan dimasukkan array dan mengembalikan JSON encode dari hasil JSON.

Class merupakan salah satu komponen dalam *library* java. *Class* harus terlebih dahulu di-*import* di awal aplikasi agar bisa digunakan. *Class* yang di-*import* sajalah yang nantinya dapat dijadikan objek dalam aplikasi. Pada aplikasi ini digunakan *class-class* standard yang dimiliki oleh Java, *Android* serta digunakan pula *class* khusus yang digunakan untuk melakukan pemanggilan *Facebook API*.

Pada sisi client menggunakan *SDK Android* yang merupakan sebuah kumpulan *tool-tool* bantuan untuk membantu *developer* menciptakan aplikasi untuk *Android*. *SDK* ini ketika ditambahkan dalam *Eclipse* akan menambahkan *environment development Android* dalam *IDE*. Sehingga *Eclipse* akan mampu mengenali komponen-komponen yang dimiliki oleh *Android*. Aplikasi ini memiliki banyak *activity* dimana setiap *activity* memiliki fungsi masing-masing sebagai client.

Class `MenuActivity.java` merupakan *class* pertama yang dipanggil ketika aplikasi dijalankan. *Class* ini memiliki tujuh menu yang bisa diakses oleh *user guest*, dua menu tambahan yang bisa diakses oleh *user member*, dan satu menu tambahan yang bisa diakses oleh *user operator*.

Pseudocode 5. Fungsi pengecekan shared preference

```
if GetSp(iduser) exist  
then AccessTokenFileTask() and  
VerifyToken()  
    If VerifyToken success  
    Then set isLogon = 1  
Else  
end
```

Pada *Activity* ini semua *user* memiliki hak akses untuk fitur *user guest*, namun harus melakukan *log in* jika mau memiliki tombol “upload resep” atau “lihat resepku”.

Setiap aplikasi terbuka pertama kali, terjadi pengecekan `id_user` terhadap *shared preference* untuk mengetahui masih adakah `id_user` dan `access_token`. Fungsi `VerifyToken()` terjadi dikarenakan adanya *asynctask* untuk mencoba mengambil data nama_user

Pseudocode 5. Fungsi Asynchronous task

```
Create progressdialog  
Set HttpClient and set HTTPGet  
Try HttpResponse and Statusline  
If statusCode is 200  
Then response.getEntity() and  
entity.getContent  
  
onPost do displayLatteRecipe()
```

Hampir semua *class* yang memiliki fungsi yang bernama `DownloadFilesTask` yang *extends* dari *AsyncTask* yang bertujuan untuk mendapatkan data dengan menjalankan fungsi `displayLatteRecipe()` dan *AsyncTask* membuat sebuah *thread* baru untuk proses itu sendiri sehingga *thread* awal tidak terganggu dan bisa jalan bersamaan. Sesudah proses eksekusi, fungsi `displayLatteRecipe()` akan dipanggil oleh *thread* untuk melakukan proses pengambilan data dari JSON serta melakukan *setting listview* dengan *class* `ResepAdapter`.

Pseudocode 6. Fungsi penyimpanan data offline

```
Set path  
Create File  
Make directory with path  
  
Create String name then  
Save file with parameter File and String  
  
Prinwriter  
Println  
Flush
```

`ViewDetailRecipe.java` merupakan *class* yang memiliki *AsyncTask* sama dengan `ViewLatestRecipeActivity.java` dengan isi data yang berbeda. Pada *class* ini terdapat beberapa fitur yaitu : *save bahan*, *save resep*, *rating*, komentar

Save bahan dan *save resep* berfungsi melakukan penyimpanan data yang sudah ditampilkan pada layar ke dalam storage *client(android device)* sehingga mampu diakses secara *offline* oleh *user*. Perbedaan kedua fungsi ini adalah pada *save bahan* yang disimpan hanya bahan saja sedangkan *save resep* akan menyimpan data *resep* secara keseluruhan.

Fitur *rating* digunakan untuk memberikan *rating* pada suatu resep makanan. Fitur *Komentar* digunakan untuk memberikan komentar pada suatu resep makanan. Proses penyimpanan data resep dimulai dari pembuatan directory untuk penyimpanan data berupa *text file*, pembuatan *file* dan penulisan data ke dalam *file* dengan menggunakan *PrintWriter*

Pseudocode 7. Fungsi penyimpanan cache gambar

```
Set path  
Create File  
Make directory with path  
  
Create String name and  
Create bitmap file then  
Save file with parameter File and String  
Try to compress bitmap to jpg, flush and  
close. Save file with mediastore
```

Proses penyimpanan gambar dengan menggunakan `buildDrawingCache()` untuk mendapatkan *cache* dari gambar yang telah ter-*download* dan menggunakan `FileOutputStream` untuk mengambil *file* gambar dan diatur sehingga masuk dalam directory yang sudah diatur.

Pseudocode 8. Fungsi pengambilan data dari text file dan decode gambar

```
Set path
If savedir exist set
savedir.isDirectory()

Try System.out.println with path.
Read each folder

Create File in FileInputStream in
InputStreamReader
decodeFile bitmap

Get data & image
```

Proses ini melakukan pengambilan data pada storage android device. Proses pengambilan data dimulai dari perhitungan jumlah *subfolder* dalam *folder* yang sudah ditentukan penyimpanannya.

Menggunakan `BufferedReader` memungkinkan untuk membaca `file.txt` perline dan dengan menggunakan `BitmapFactory.decodeFile` untuk melakukan decode image yang ada pada storage

Pseudocode 9. Fungsi Log in dengan email dan password

```
Encode username
Encode password

HttpRequest with username and password

If username and password valid then
setsp()
```

Proses ini melakukan pengiriman data username dan password kepada webserver dan webserver melakukan pengecekan. Ketika data valid maka username melakukan setting `access_token` dan `access token` akan hilang pada saat melakukan logout.

Pseudocode 10. Fungsi Log in dengan facebook

```
Fb.authorize()
If Fb.authorize() success
Then
Fb.getAccessToken()
Fb.getAccessExpires()
Editor.commit()
```

Proses melakukan log in menggunakan facebook, user diminta untuk pastinya memiliki aplikasi dan account facebook terlebih dahulu. Proses *Log in* akan muncul ketika button facebook log in mengalami *OnClick* sehingga muncul halaman facebook yang meminta permission user untuk menggunakan account facebook dalam aplikasi ini. Pada saat berhasil log in, status user memiliki `access_token` dan `access_expires` dan disimpan pada `shared preference`.

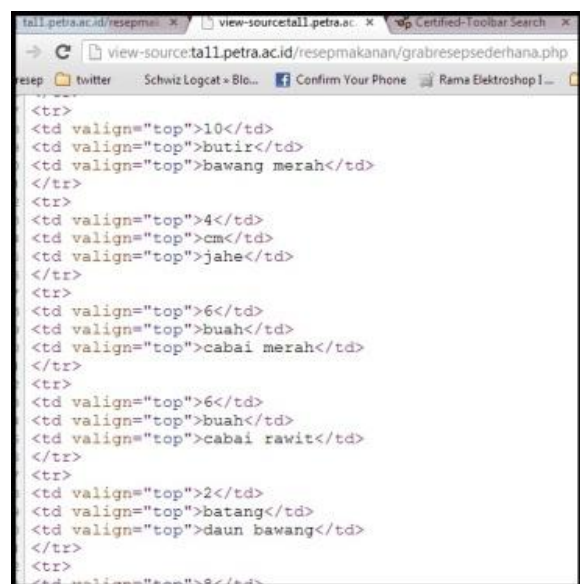
5. PENGUJIAN SISTEM

Pengujian pertama akan dimulai dengan data mentah yang telah disediakan dalam RSS pada website yang telah disediakan sampai dengan data yang siap pakai dalam database.



Gambar 1. Halaman RSS suatu website

Pada **Gambar 1** merupakan tahap pertama pengujian system yang menunjukkan bahwa website menyediakan RSS untuk dapat diolah oleh pengguna web dan dapat dimanfaatkan oleh banyak developer.



Gambar 2. Halaman hasil YQL melalui source code

Pada **Error! Reference source not found.** menunjukkan hasil dari pembacaan RSS dengan YQL sehingga data yang ada dapat dicari polanya untuk dapat dipisah dan dimasukkan kedalam database.

id_bahan	id_nama_makanan	bahan	jumlah	satuan	nama_bahan	ket
1	78	25 gram wijen putih	25	gram	wijen putih	
2	78	25 gram wijen hitam	25	gram	wijen hitam	
3	78	75 gram margarin	75	gram	margarin	
4	78	70 gram gula palem	70	gram	gula palem	
5	78	1 butir kuning telur	1	butir	kuning telur	
6	78	2 sdm susu kental manis	2	sdm	susu kental manis	
7	78	170 gram tepung kaji, disangrai	170	gram	tepung kaji	disangrai
8	78	2 butir putih telur, kocok lepas	2	butir	putih telur	kocok lepas
9	78	1 sdt untuk olesan loyang	1	sdm	untuk olesan loyang	
10	79	20 gram gula halus	20	gram	gula halus	
11	79	90 gram mentega	90	gram	mentega	
12	79	1 butir kuning telur	1	butir	kuning telur	
13	79	100 gram kenari bubuk	100	gram	kenari bubuk	
14	79	2 sdm selai bluberi	2	sdm	selai bluberi	
15	79	150 gram tepung terigu protein rendah	150	gram	tepung terigu protein rendah	

Gambar 3.Hasil pemotongan data yang dimasukkan dalam MySql

Pada **Error! Reference source not found.** menunjukkan data yang sudah siap pakai untuk memberi data kepada *client*.Bahan dipecah menjadi tiga bagian yang berfungsi untuk mendukung fitur *filtering* yang sudah ditentukan.

```
String parse
{
    "id_nama_makananan":"136",
    "nama_makananan":"Nasi Tim Telur Udang",
    "bahan": [
        {
            "bahan": "4 siung bawang putih"
        },
        {
            "bahan": "1\2 buah bawang bombay"
        },
        {
            "bahan": "2 batang daun bawang"
        },
        {
            "bahan": "400 gram udang"
        },
        {
            "bahan": "200 gram kacang polong"
        },
        {
            "bahan": "4 butir telur ayam"
        },
        {
            "bahan": "400 gram beras"
        },
        {
            "bahan": "1 sdt tepung kanji"
        }
    ]
}
```

Gambar 4.Format pengiriman data

Pada **Error! Reference source not found.** merupakan format data yang dikirim *webserver* kepada *client* untuk setiap permintaan *client* terhadap *webserver*.

6. KESIMPULAN

Kesimpulannya adalah aplikasi telah berhasil melakukan pengambilan data dari source untuk dimasukkan kedalam webserver, mengirimkan data dari client kepada webserver, hingga pengiriman data dari webserver kepada client.

Dalam kasus pengkodean, pengkodean menggunakan platform Android 2.2 dan diuji menggunakan OS 4.0 memunculkan *force close* pada Android OS 4.0, khususnya dalam hal pengiriman data atau *request* data yang pada Android 2.2 tidak memerlukan *AsyncTask* dan Android OS 4.0 memerlukan *AsyncTask*.

7. DAFTAR PUSTAKA

- [1] Android. (2006). *Software Development Kit*.retrieved September, 19, 2012, from <http://developer.android.com>.
- [2] Apache. (2004). *Apache webserver*. retrieved August, 17, 2012 from <http://www.apache.org/>
- [3] Bodmer, Fabrice (2007), *Cross-Site Scripting(XSS)Computer and Network Security Seminar*.Maryland: Syngress.
- [4] Crontab. (2008). *Crontab Scheduling*. Retrieved October, 25, 2012, From <http://www.adminschoice.com/crontab-quick-reference>.
- [5] D. Crockford(2006), *Network Working Group* . Washington : UnitedStates Patent.
- [6] JSON, (2009). *JSON format*. Retrieved August, 17, 2012, from <http://www.json.org>.
- [7] Rogers,Rick., Lombardo, John., Mednieks, Zigurd., Blake Meike(2009), *Android Application Development: Programming with the Google SDK1st*. California : O'Reilly Media, Inc
- [8] Yahoo. (2006). *Yahoo Query Language read RSS*. Retrieved August, 25, 2012, from <http://developer.yahoo.com/yql>